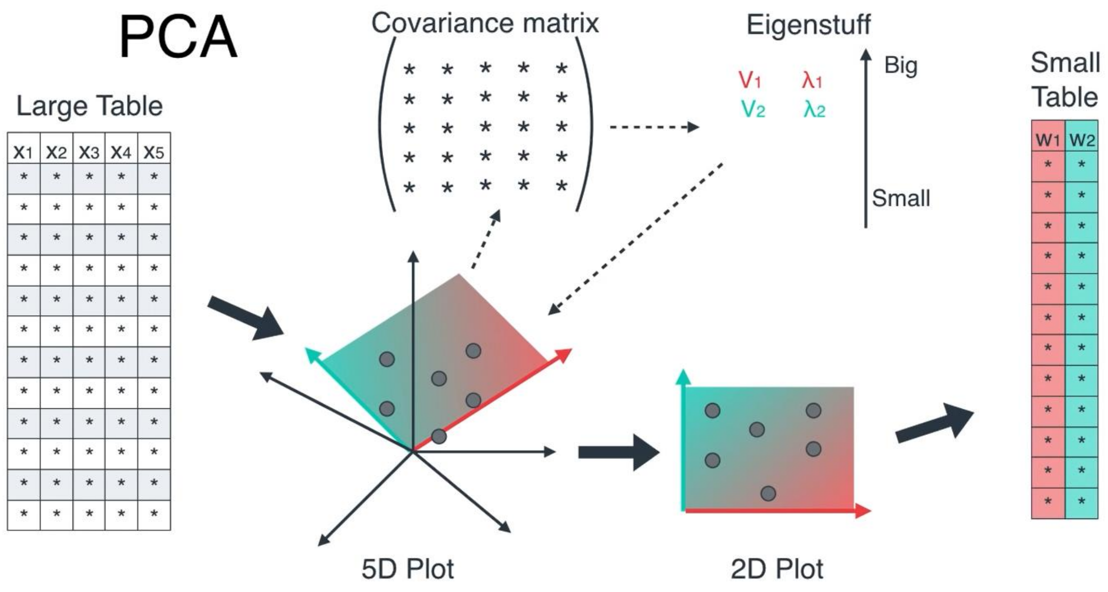
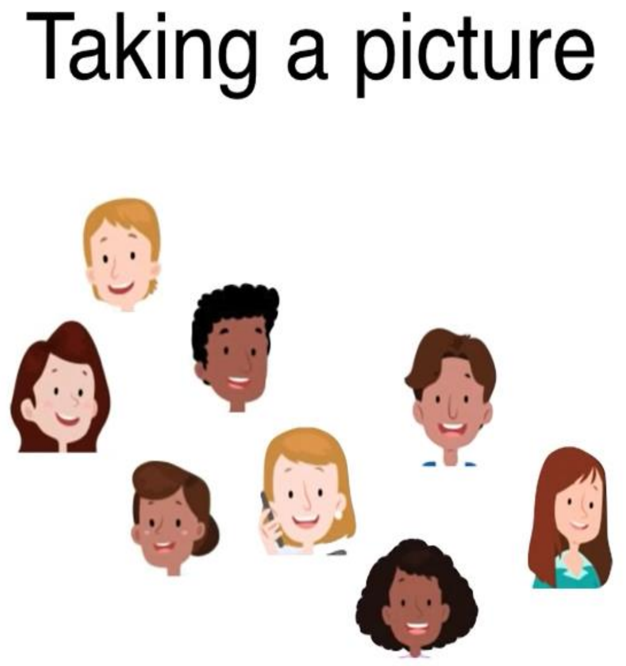
Principal Component Analysis

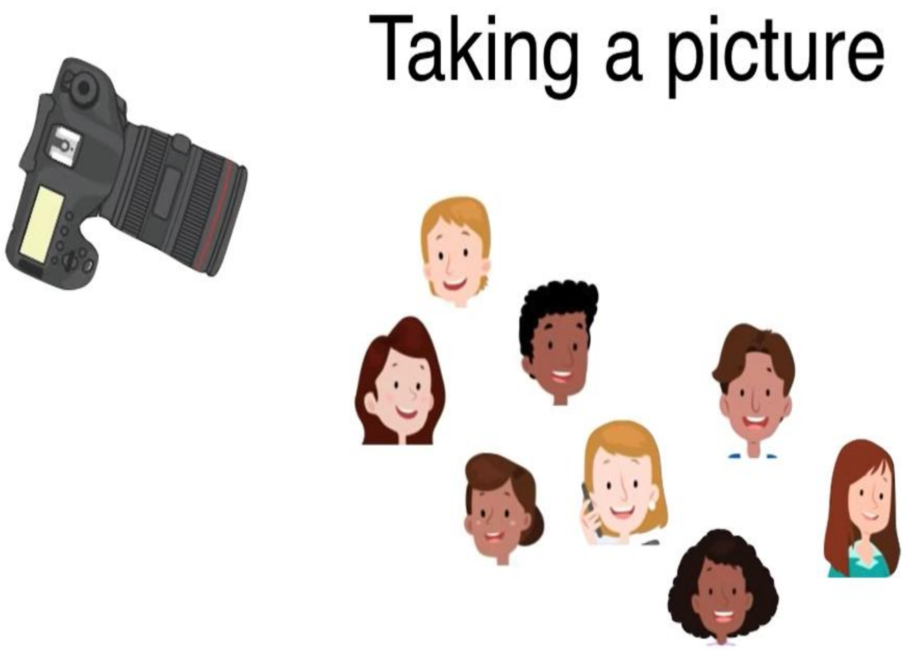
Data Set: **pca3.xlsx**



From which angle you can capture maximum information (Variation/Variance)

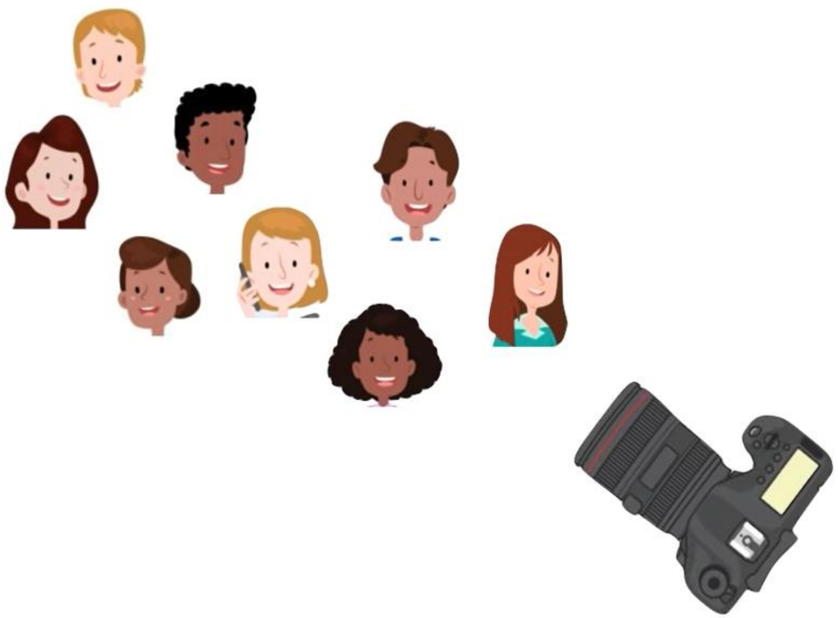


Is this a good angle/projection?



Is this a good angle/projection?

Taking a picture

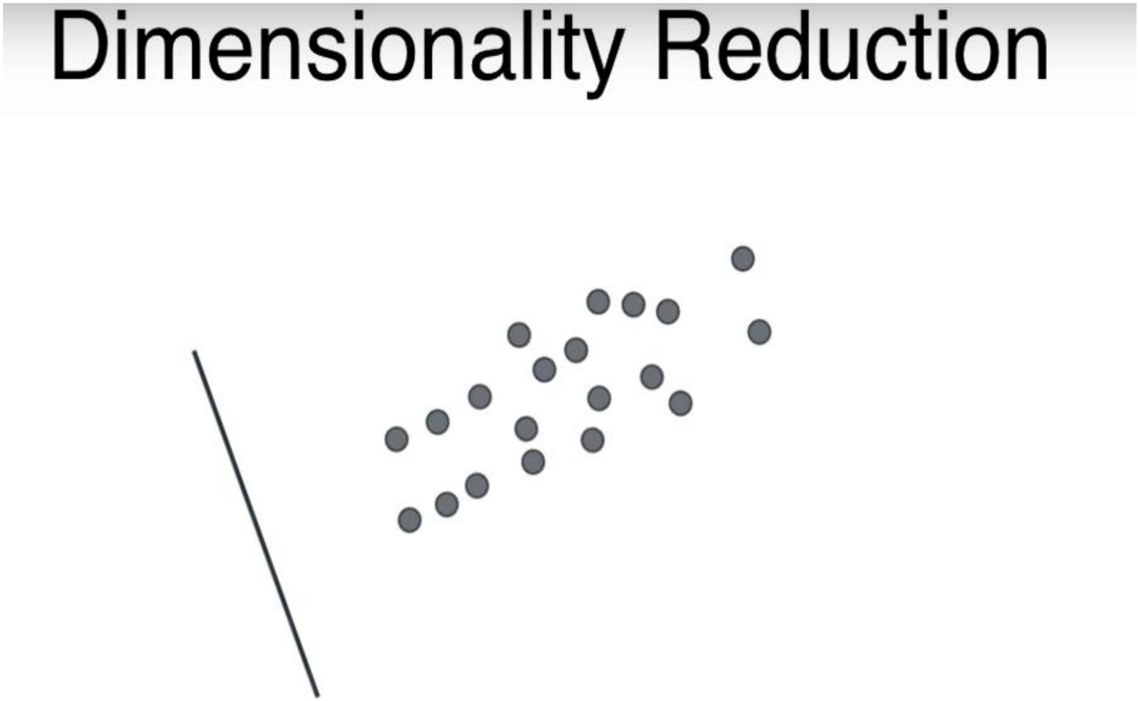


Is this a good angle/projection?

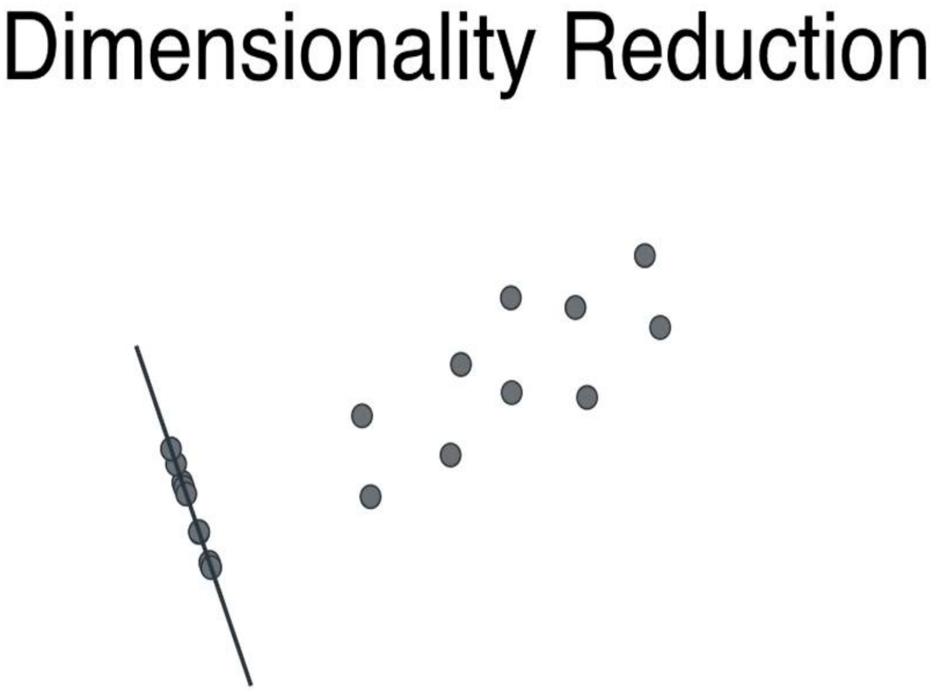
Look at this data and try to find right/proper/eigen projection

Dimensionality Reduction

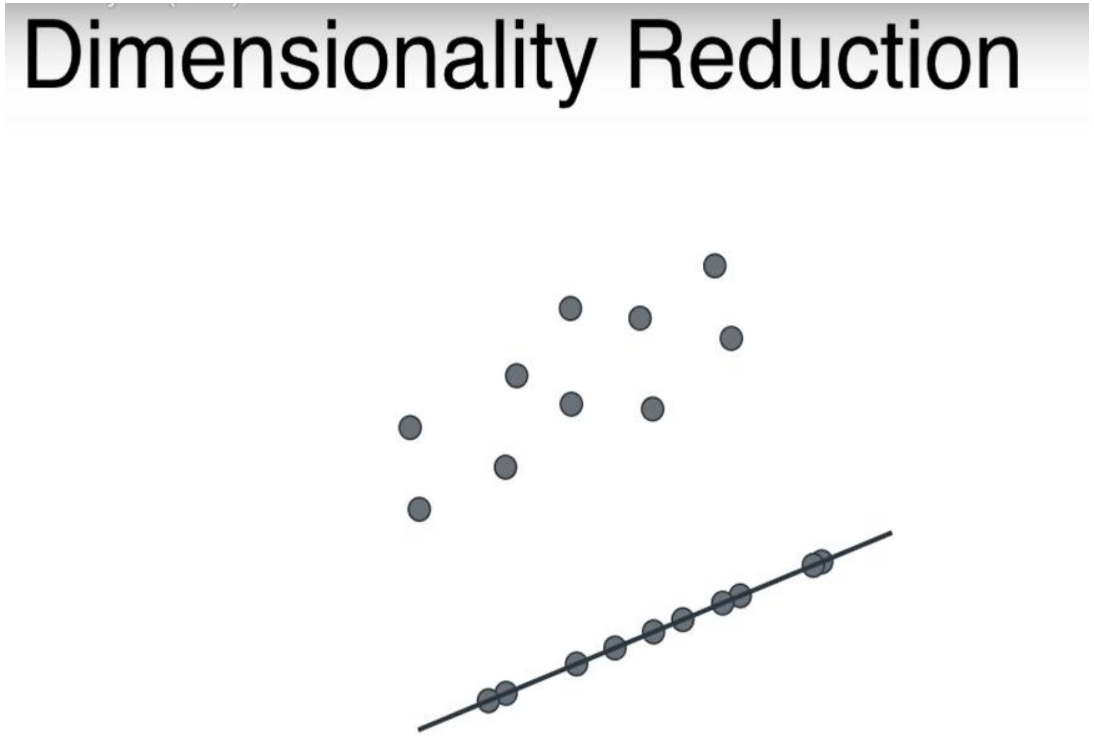
Can this be a good projection?



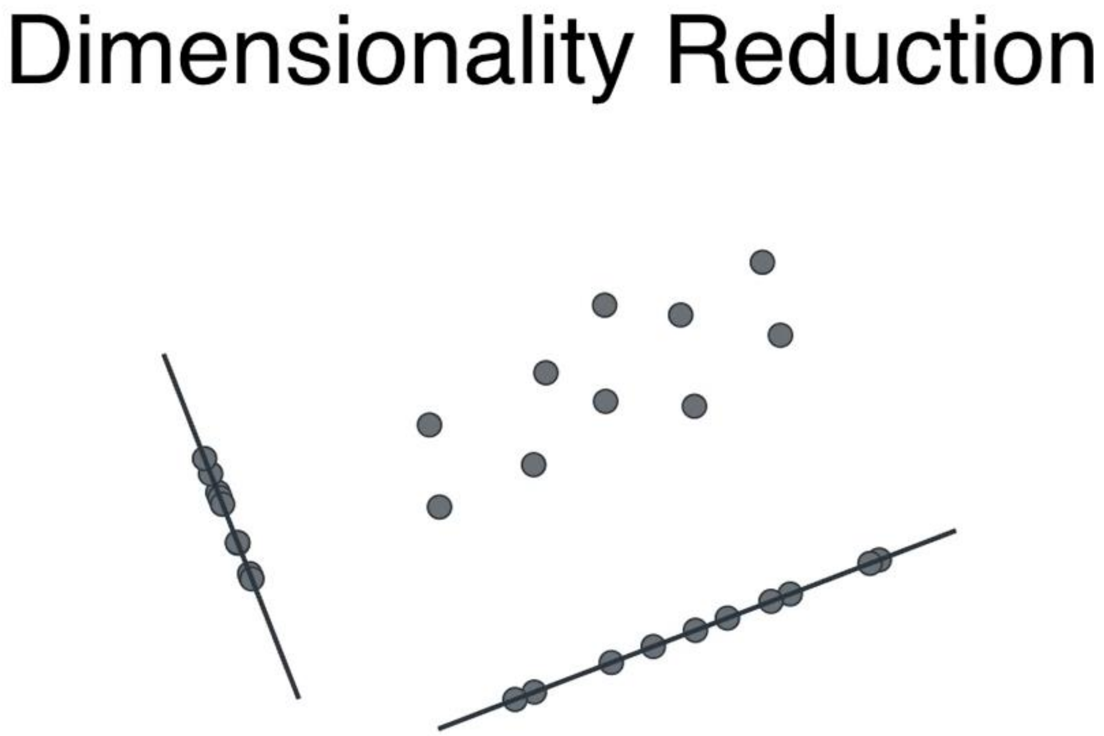
Can this be a good projection?



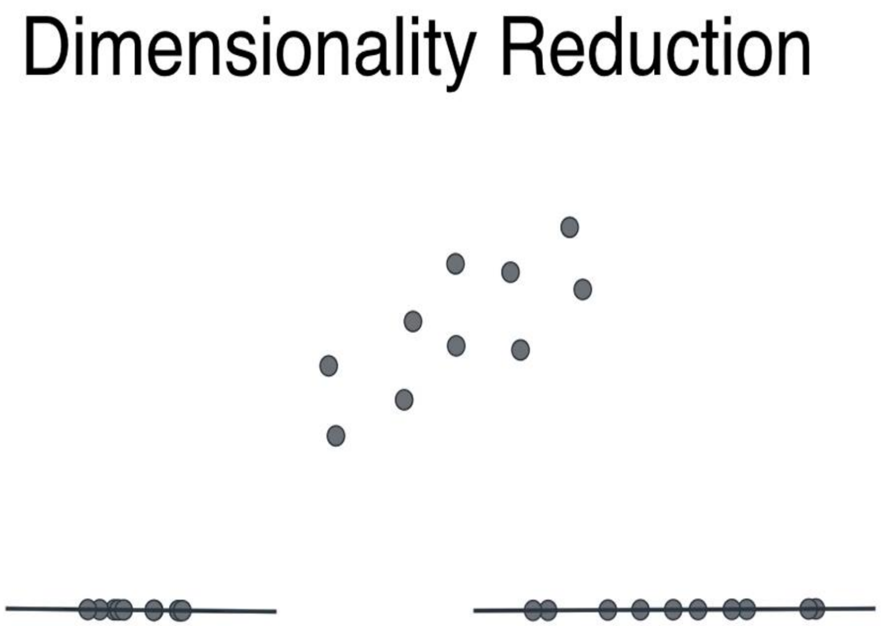
Can this be a good projection?



Out of these 2 projections, which one is good and why?

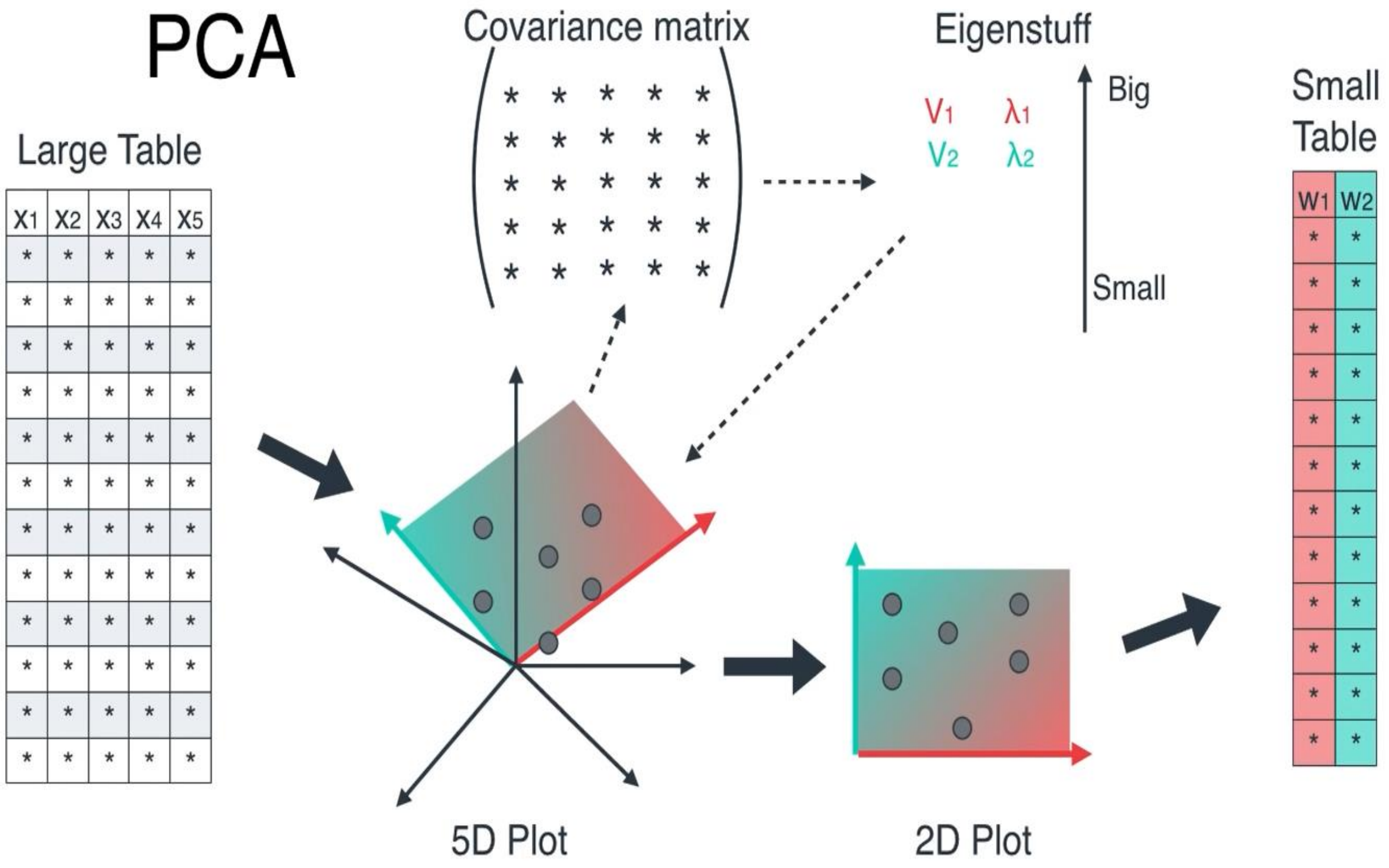


Out of these 2 projections, which one is good and why?



Out of these 2 projections, which one is good and why?

Procedure

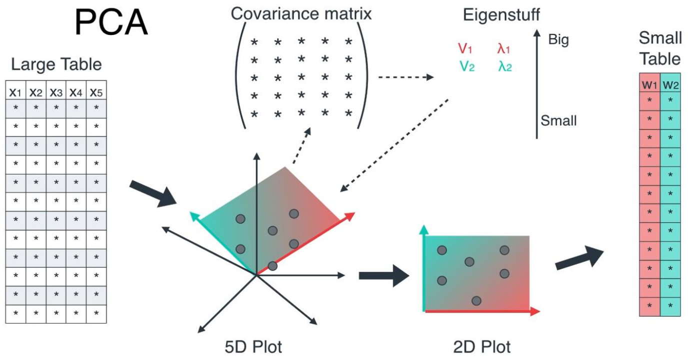

PCA
Large Table
X1 X2 X3 X4 X5
Covariance matrix
Eigenstuff
Big
V1 λ1
V2 λ2
Small
Small Table
W1 W2
5D Plot
2D Plot

Data

```r
# Jesus is my Savior!
library(readxl)
pca3 <- read_excel("C:/Users/Dr Vinod/Desktop/DataSets1/pca3.xlsx")
#View(pca)
pca3 <- as.data.frame(pca3)
str(pca3)
```

```
> str(pca3)
'data.frame':	10 obs. of  3 variables:
 $ v1: num  7 4 6 8 8 7 5 9 7 8
 $ v2: num  4 1 3 6 5 2 3 5 4 2
 $ v3: num  3 8 5 1 7 9 3 8 5 2
```

	A	B	C
1	v1	v2	v3
2	7	4	3
3	4	1	8
4	6	3	5
5	8	6	1
6	8	5	7
7	7	2	9
8	5	3	3
9	9	5	8
10	7	4	5
11	8	2	2

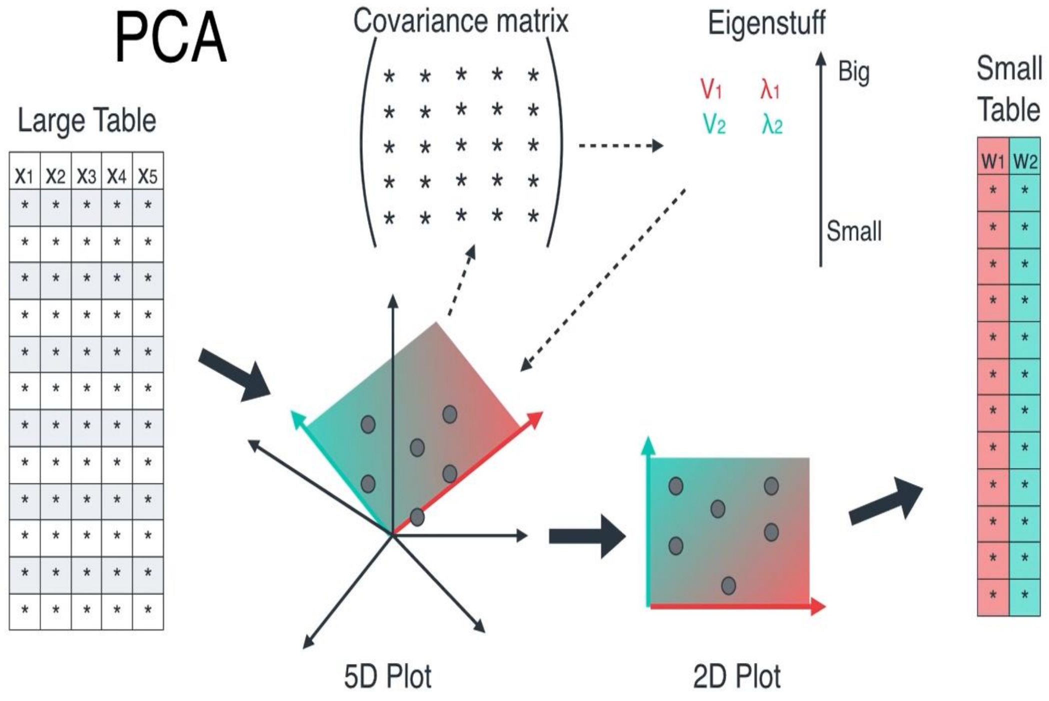


fdp.R | pca3 | pca3.R | tdata

Filter

	v1	v2	v3
1	7	4	3
2	4	1	8
3	6	3	5
4	8	6	1
5	8	5	7
6	7	2	9
7	5	3	3
8	9	5	8
9	7	4	5
10	8	2	2

Mathematical Treatment



Step 1: Centering
Step 2: Covariance Matrix
Step 3: Use Characteristic Equation det of ($A- \lambda I = 0$) for setting up equation for unknown λ (eigen values)
Step 4: Find eigen vectors
Step 5: Transform centered data by Matrix Multiplication. [M1 = centered data (10by3); M2 = Eigen Vectors (3by3)

Eigen Values

$$A = \begin{bmatrix} 8 & -6 & 2 \\ -6 & 7 & -4 \\ 2 & -4 & 3 \end{bmatrix}$$

characteristic eqn:

$$\det.\left[A - \lambda I\right] = 0$$

First find $[A - \lambda I]$

$$\begin{bmatrix} 8 & -6 & 2 \\ -6 & 7 & -4 \\ 2 & -4 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} (8-\lambda) & -6 & 2 \\ -6 & (7-\lambda) & -4 \\ 2 & -4 & (3-\lambda) \end{bmatrix}$$

det. of above: (i) follow +, −, +
(ii) start from NW corner (or can go down ☺)
(iii) hide that row & column

$$+ (8-\lambda)\left[(7-\lambda)\times(3-\lambda) - (-4\times -4)\right] - (-6)\left[(-6)\times(3-\lambda) - (-4\times 2)\right] + 2\left[(-6)\times(-4) - 2\times(7-\lambda)\right]$$

$$\boxed{\lambda^3 - 18\lambda^2 + 45\lambda = 0}$$

$\lambda = 0, \lambda = 3, \lambda = 15$

for $\lambda = 0$ eigen vector PC3

$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$ ☺

Eigen Vectors

This is same as λI

For $\lambda = 0$

$$[A - \lambda][x] = 0$$

$$\begin{bmatrix} (8-\lambda) & -6 & 2 \\ -6 & (7-\lambda) & -4 \\ 2 & -4 & 3-\lambda \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Keeping $\lambda = 0$

$$\begin{bmatrix} 8 & -6 & 2 \\ -6 & 7 & -4 \\ 2 & -4 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$8x - 6y + 2z = 0 \quad -①$$
$$-6x + 7y - 4z = 0 \quad -②$$
$$2x - 4y + 3z = 0 \quad -③$$

Let's solve ① & ② You can take any two ☺
(+, −, + applicable)

$$+\frac{x}{(-6)\times(-4) - (7\times 2)} - \frac{y}{8\times(-4) - (-6)\times 2} + \frac{z}{8\times 7 - [-6\times(-6)]}$$

$$\frac{x}{24-14} - \frac{y}{-32-(-12)} + \frac{z}{56-36}$$

$$\frac{x}{10} - \frac{y}{-20} + \frac{z}{20}$$

Divide by 10

$$\boxed{\frac{x}{1} + \frac{y}{2} + \frac{z}{2}}$$

Matrix Multiplication

SUM =G7*L4+H7*L5+I7*L6

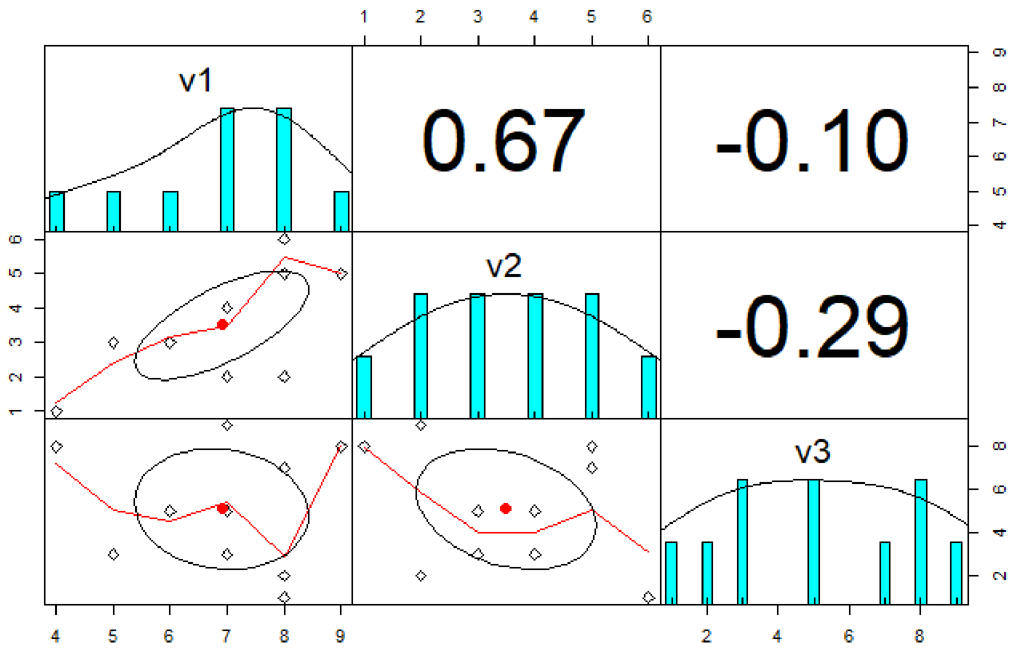
	F	G	H	I	J	K	L	M	N	O	P	Q
1		**Centered Data 10by3**			**Eigen Vectors 3by3**					**a_{ij} (i=row in M1, j=col in M2)**		
2						lambda1	lambda2	lambda3		**1**	**2**	**3**
3		v1c	v2c	v3c	**eigen values**	**8.2761**	**3.6747**	**0.7493**		PC1	PC2	PC3
4	**1**	0.1	0.5	-2.1	v1	**-0.1369**	**0.6991**	**-0.7018**	**1**	-2.15158	-0.17226	0.10581
5	**2**	-2.9	-2.5	2.9	v2	**-0.2505**	**0.661**	**0.7073**	**2**	3.80262	-2.88906	0.51231
6	**3**	-0.9	-0.5	-0.1	v3	**0.9584**	**0.2727**	**0.0846**	**3**	0.15262	-0.98696	0.26951
7	**4**	1.1	2.5	-4.1					**4**	-4.70628	=G7*L4	0.64941
8	**5**	1.1	1.5	1.9					**5**	1.29462	2.27864	0.44971
9	**6**	0.1	-1.5	3.9					**6**	4.09982	0.14194	-0.80119
10	**7**	-1.9	-0.5	-2.1					**7**	-1.62728	-2.23146	0.80211
11	**8**	2.1	1.5	2.9					**8**	2.11612	3.25044	-0.16749
12	**9**	0.1	0.5	-0.1					**9**	-0.23478	0.37314	0.27501
13	**10**	1.1	-1.5	-3.1					**10**	-2.74588	-1.06786	-2.09519
14		0	0	0						4.44089E-16	0	0
15		**1.523884**	**1.581139**	**2.806738**						**2.876477926**	**1.91735**	**0.865965**

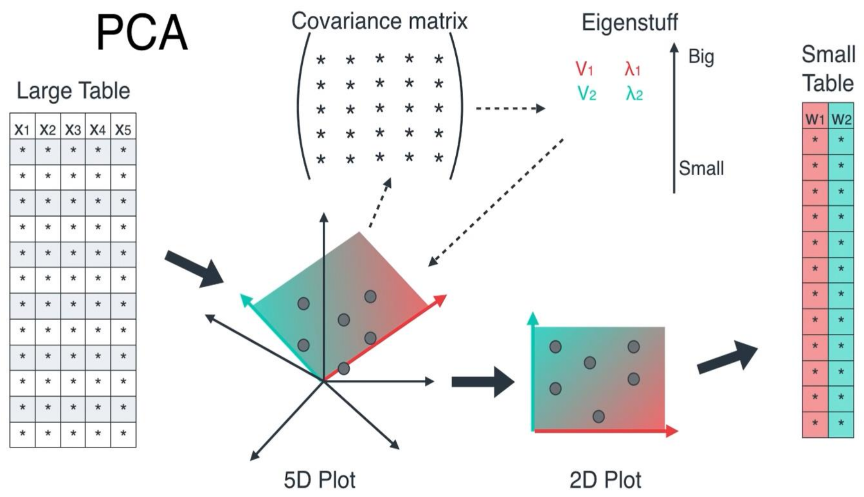
```
> print(pca)
Standard deviations (1, .., p=3):
[1] 2.8764462 1.9173235 0.8659839

Rotation (n x k) = (3 x 3):
          PC1       PC2         PC3
v1 -0.1375708 0.6990371 -0.70172743
v2 -0.2504597 0.6608892  0.70745703
v3  0.9583028 0.2730799  0.08416157
```

Visualize Raw Data

```
library(psych)
pairs.panels(pca3,
             gap = 0, pch = 5)
```

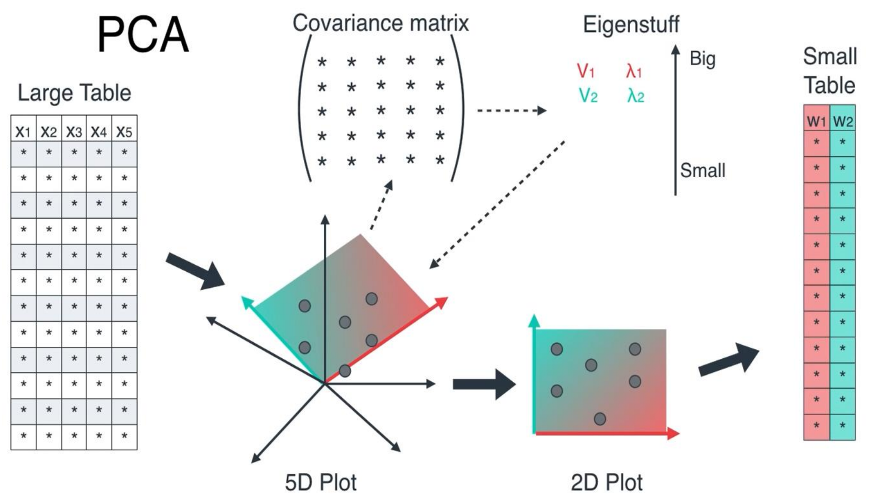




Execute and see what is stored in output

```
pca <- prcomp(pca3,
              center = TRUE,
              scale = FALSE)

attributes(pca)
```

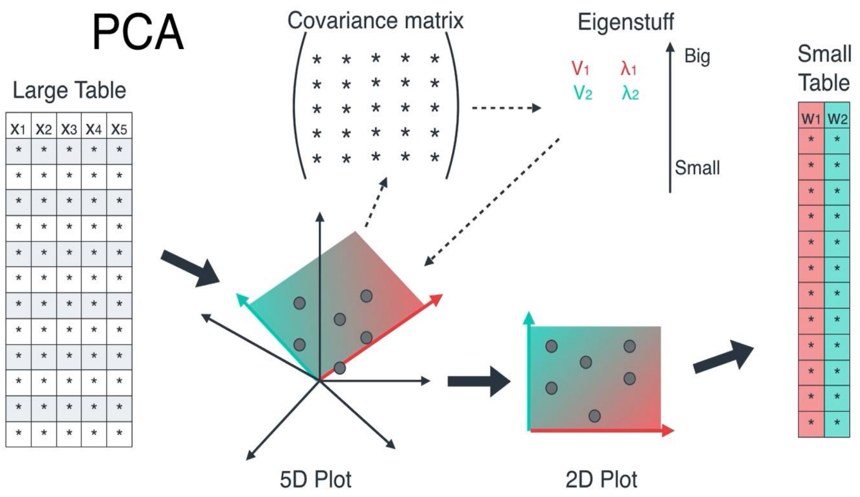



```
> attributes(pca)
$`names`
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

Standard Deviation of Transformed Data

```
> pca$sdev # [1] 2.8764  1.9173  0.8659
[1] 2.8764462 1.9173235 0.8659839
```

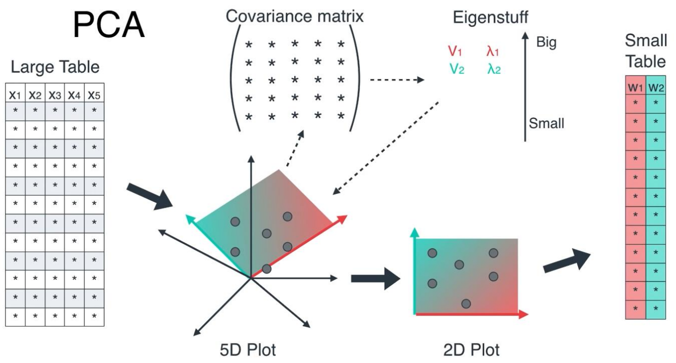


	A	B	C	D	E
1	**Transformed data 10by3**				
2	eigen values =	**8.2761**	**3.6747**	**0.7493**	
3		PC1	PC2	PC3	
4	1	-2.15142	-0.17312	0.106816	
5	2	3.804183	-2.8875	0.510436	
6	3	0.153213	-0.98689	0.26941	
7	4	-4.70652	1.301536	0.65168	
8	5	1.293758	2.279126	0.449192	
9	6	4.099313	0.143581	-0.80313	
10	7	-1.62582	-2.23208	0.802814	
11	8	2.11449	3.251243	-0.16837	
12	9	-0.23482	0.37304	0.27514	
13	10	-2.74638	-1.06894	-2.09399	
14	avg=	1.78E-15	7.33E-16	0	
15	**sd=**	**2.876446**	**1.917323**	**0.865984**	
16					
17	eigen values	**8.2761**	**3.6747**	**0.7493**	**12.7001**
18		0.651656	0.289344	0.059	
19			0.941	1	

Eigen Vectors extracted and Centers (averages of Raw Data)

```
> pca$rotation # PC1:PC3 as columns
          PC1       PC2         PC3
v1 -0.1375708 0.6990371 -0.70172743
v2 -0.2504597 0.6608892  0.70745703
v3  0.9583028 0.2730799  0.08416157
```

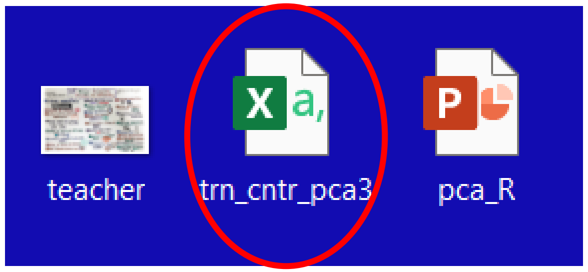
```
> pca$center # 3 cntrs, Raw Averages
 v1  v2  v3
6.9 3.5 5.1
```

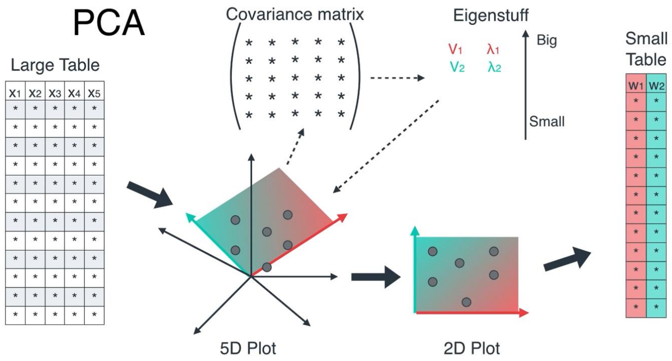


	A	B	C	D
1				
2		**Raw Data**		
3		v1	v2	v3
4	1	7	4	3
5	2	4	1	8
6	3	6	3	5
7	4	8	6	1
8	5	8	5	7
9	6	7	2	9
10	7	5	3	3
11	8	9	5	8
12	9	7	4	5
13	10	8	2	2
14	**Avg =**	**6.9**	**3.5**	**5.1**
15	sd =	1.523884	1.581139	2.8067379

You can have transformed data at desktop for further experiments!

```
tdata_cntr = as.data.frame(pca$x)
write.csv(tdata_cntr, "C:/Users/Dr Vinod/Desktop/trn_cntr_pca3.csv")
```



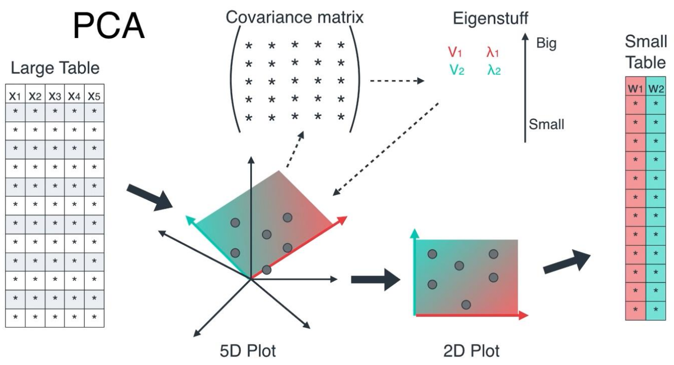


	A	B	C	D	E
1	**Transformed data 10by3**				
2	eigen values =	**8.2761**	**3.6747**	**0.7493**	
3		PC1	PC2	PC3	
4	1	-2.15142	-0.17312	0.106816	
5	2	3.804183	-2.8875	0.510436	
6	3	0.153213	-0.98689	0.26941	
7	4	-4.70652	1.301536	0.65168	
8	5	1.293758	2.279126	0.449192	
9	6	4.099313	0.143581	-0.80313	
10	7	-1.62582	-2.23208	0.802814	
11	8	2.11449	3.251243	-0.16837	
12	9	-0.23482	0.37304	0.27514	
13	10	-2.74638	-1.06894	-2.09399	
14	avg=	1.78E-15	7.33E-16	0	
15	**sd=**	**2.876446**	**1.917323**	**0.865984**	
16					
17	eigen values	**8.2761**	**3.6747**	**0.7493**	**12.7001**
18		0.651656	0.289344	0.059	
19			0.941	1	

SDs of Transformed Data & Eigen Vectors

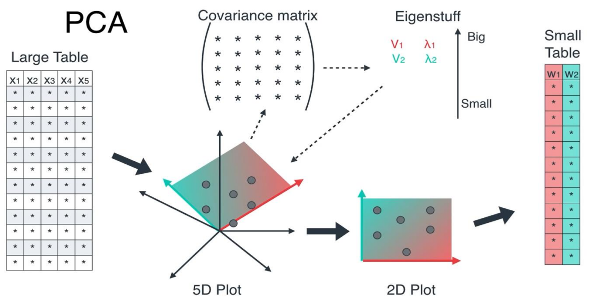
```
> print(pca)
Standard deviations (1, .., p=3):
[1] 2.8764462 1.9173235 0.8659839

Rotation (n x k) = (3 x 3):
          PC1       PC2         PC3
v1 -0.1375708 0.6990371 -0.70172743
v2 -0.2504597 0.6608892  0.70745703
v3  0.9583028 0.2730799  0.08416157
```



	A	B	C	D	E
1	**Transformed data 10by3**				
2	eigen values =	**8.2761**	**3.6747**	**0.7493**	
3		PC1	PC2	PC3	
4	1	-2.15142	-0.17312	0.106816	
5	2	3.804183	-2.8875	0.510436	
6	3	0.153213	-0.98689	0.26941	
7	4	-4.70652	1.301536	0.65168	
8	5	1.293758	2.279126	0.449192	
9	6	4.099313	0.143581	-0.80313	
10	7	-1.62582	-2.23208	0.802814	
11	8	2.11449	3.251243	-0.16837	
12	9	-0.23482	0.37304	0.27514	
13	10	-2.74638	-1.06894	-2.09399	
14	avg=	1.78E-15	7.33E-16	0	
15	**sd=**	**2.876446**	**1.917323**	**0.865984**	
16					
17	eigen values	**8.2761**	**3.6747**	**0.7493**	**12.7001**
18		0.651656	0.289344	0.059	
19			0.941	1	

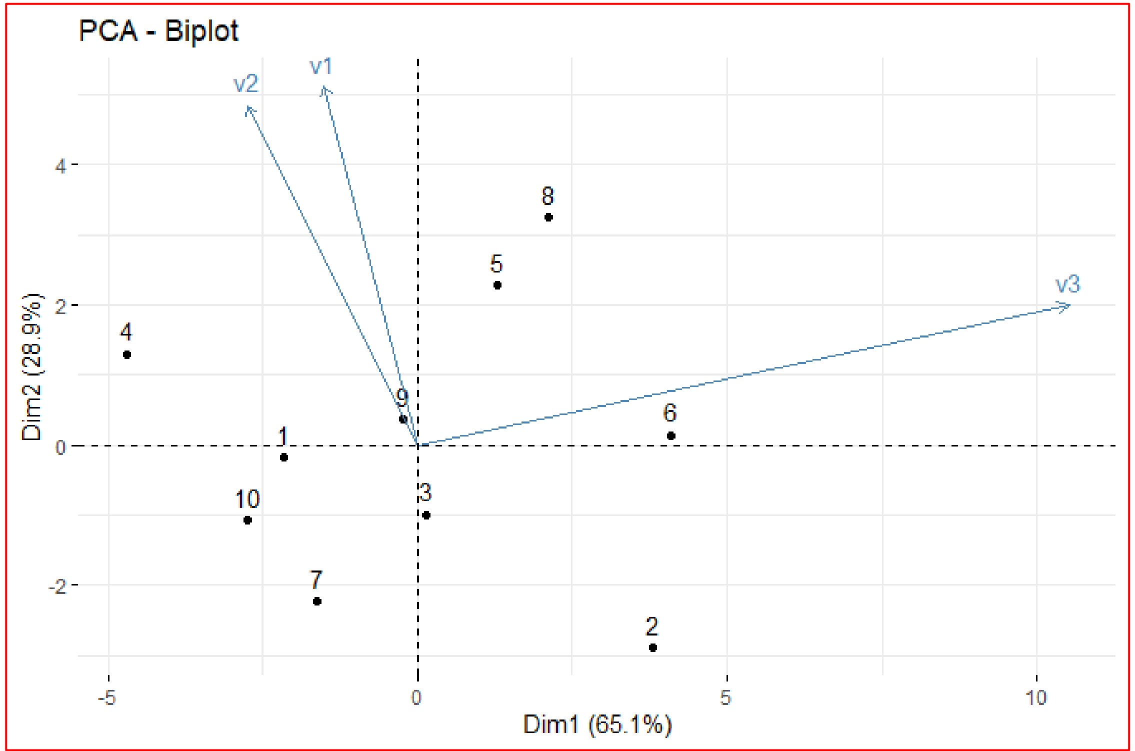
Summary: PCA




```
> summary(pca) # 3 PCs, sds, props, cum prop
Importance of components:
                          PC1    PC2     PC3
Standard deviation     2.8764 1.9173 0.86598
Proportion of Variance 0.6515 0.2895 0.05905
Cumulative Proportion  0.6515 0.9409 1.00000
```

```
library(factoextra)
res.pca<- prcomp(pca3, center = TRUE, scale = FALSE)
fviz_pca_biplot(res.pca)
```

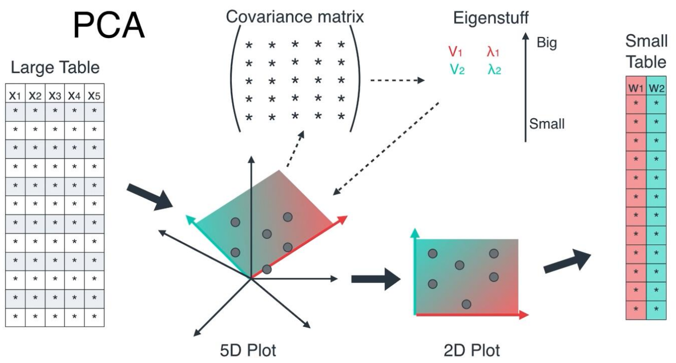
	A	B	C	D	E
1	**Transformed data 10by3**				
2	eigen values =	**8.2761**	**3.6747**	**0.7493**	
3		PC1	PC2	PC3	
17	eigen values	**8.2761**	**3.6747**	**0.7493**	**12.7001**
18		0.651656	0.289344	0.059	
19			0.941	1	

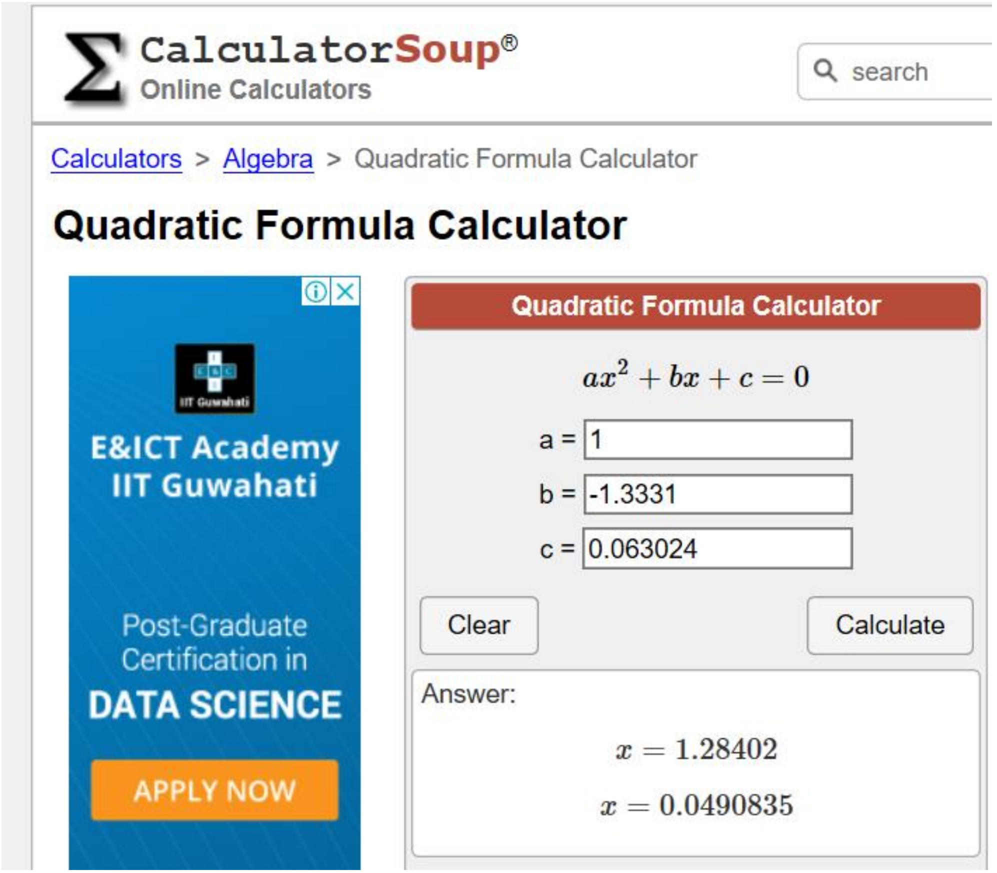


Calculation: No Worries!

Taking a picture

https://www.calculatorsoup.com/calculators/algebra/quadratic-formula-calculator.php, last accessed 10 November 2019






Happy
Learning